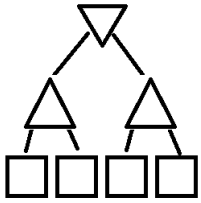
Recitation 5 (Midterm Review) February 14 ♡

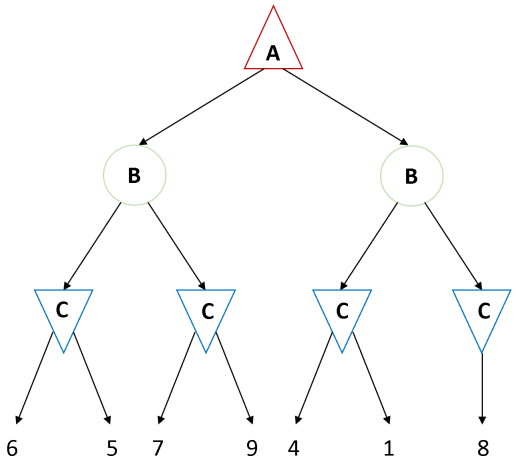
1 Adversarial Search (Minimax+Expectimax Pruning)

1. Consider the following generic tree, where the triangle pointing down is a minimizer, the triangles pointing up are maximizers, and the square leaf nodes are terminal states with some value that has not been assigned yet:

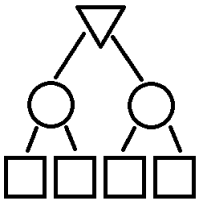


For each leaf node, is it possible for that leaf node to be the leftmost leaf node pruned? If yes, give an example of terminal values that would cause that leaf to be pruned. If not, explain. How might your answers change if the total range of the values were fixed and known?

2. Recall that a chance node has the expected value of its children, and let each child have an equal probability of being chosen.
Perform pruning on the following game tree and fill in the values at letter nodes. A is a maximizer, B is a chance node, and C is a minimizer. Assume that values can only be in the range 0-9 (inclusive).



3. Now repeat the exercise from question 1 with the following tree, where the maximizer nodes have been replaced with expectimax nodes. Assume the range of values is unknown. How would your answer change if the range of values were fixed and known (say, between 0-9)?



2 CSPs

You've generously saved a row of 6 seats in Rashid for 6 of your 15-281 classmates (A-F[1]), and are now trying to figure out where each person will be seated. You know the following pairs of people have some kind of binary constraint between them:

- A, B
- A, C
- A, D
- B, C
- B, E
- C, F

1. Draw the constraint graph to represent this CSP.

2. Some value is assigned to A. Which domains could change as a result of running forward checking for A?

3. Some value is assigned to A, and then forward checking is run for A. Then some value is assigned to B. Which domains could change as a result of running forward checking for B?

4. Some value is assigned to A. Which domains could change as a result of running AC-3 after this assignment?

5. Some value is assigned to A, and then arc consistency is enforced via AC-3. Then some value is assigned to B. Which domains will and will not change as a result of enforcing arc consistency after B's assignment?

6. Suppose the actual constraints of this CSP are as follows:

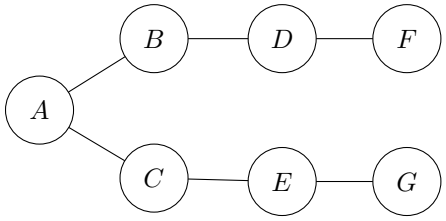
   Both C and E want to sit next to B.
   A wants to sit next to D, but not next to B or C.
   F and C had a falling out over whether AI or blockchain was cooler, so there needs to be at least 2 seats between them.

   B gets to class first and sits down in seat 3. Run AC-3 to determine the final seating arrangement (more space on next page to work it out).

[1] not correlated with their grades

7. You are now trying a brand new algorithm to solving CSPs by enforcing arc consistency via AC-3 initially, then **after every even-numbered assignment** of variables (after assigning 2 variables, then after 4, etc.).

   You have to backtrack if, after assigning a value to variable X, there are no constraint-satisfying solutions. For a single variable with d values remaining, it is possible to backtrack $d-1$ times in the worst case. For the following constraint graph, assume each variable has a domain of size d. How many times would you have to backtrack in the worst case for the specified orderings of assignments?



   (a) ABCDEFG:

   (b) GECABDF:

   (c) GFEDCBA:

3 Search (Algorithms & Properties)

1. When can we guarantee completeness and optimality (if ever) for each of the following search algorithms we've seen in class? For each algorithm, indicate under what conditions it is complete and/or optimal.

Algorithm	**Complete**	**Optimal**
Breadth-First		
Depth-First		
Iterative Deepening		
A*		

2. Consider a dynamic A* search in which after running A* graph search and finding an optimal path from start to goal (assuming there's only one goal), the cost of one of the edges $X \to Y$ in the graph changes. Instead of re-running the entire search, you want to find a more efficient way of returning the optimal path for this new search problem.

   For each of the following changes, describe how the optimal path cost would change (if at all). If the optimal path itself changes, describe how to find the new optimal path. Denote c as the original cost of $X \to Y$, and assume $n > 0$.

   (a) c is increased by n, $X \to Y$ is on the optimal path, and X was explored by the initial search.

   (b) c is decreased by n, $X \to Y$ is on the optimal path, and X was explored by the initial search.

   (c) c is increased by n, $X \to Y$ is not on the optimal path, and X was explored by the initial search.

   (d) c is decreased by n, $X \to Y$ is not on the optimal path, and X was explored by the initial search.

   (e) c is increased by n, $X \to Y$ is not on the optimal path, and was not explored by the initial search.

   (f) c is decreased by n, $X \to Y$ is not on the optimal path, and was not explored by the initial search (assuming the edge weights c can't go negative.)

4 Local Search

4.1 Warm Up

1. Define completeness and optimality for local search problems.

2. What are two advantages of local search?

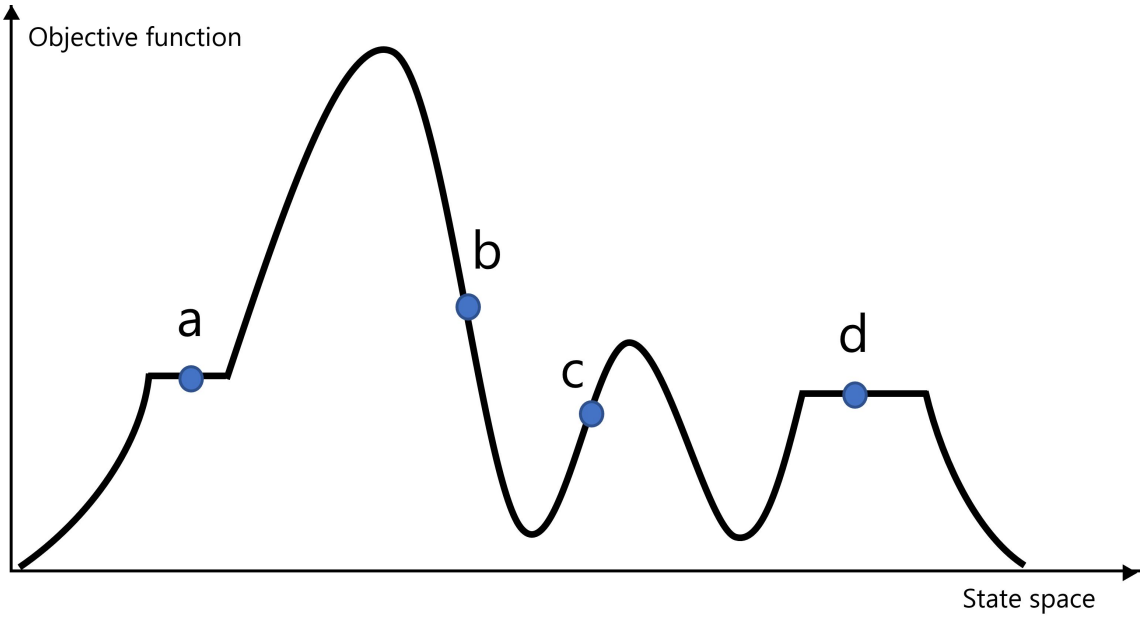
3. What are two disadvantages of local search techniques?

4. What are four functions used in genetic algorithms?

5. What two local search techniques does simulated annealing combine?

6. What is one disadvantage of beam search and how can we resolve it?

4.2 Practice



Consider how each of the following searches performs in state space above. Recall that in the context of local search, our goal is to find the state that optimizes the objective function.

1. Hill-climbing search with start state c

   Does it terminate? If so, where?

   Does it find the global maximum?

2. Random-restart hill climbing with randomly generated initial states a, d, then b

   Does it terminate? If so, where?

   Does it find the global maximum?

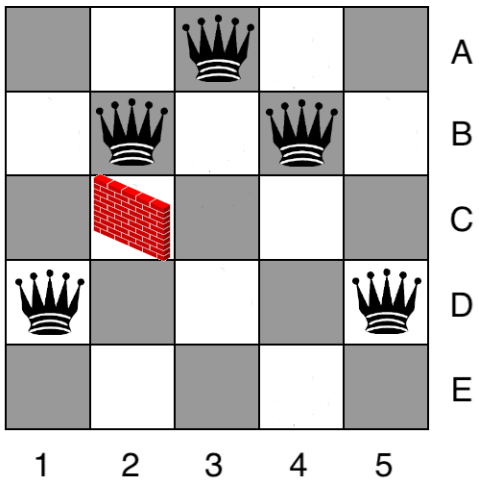
3. Stochastic hill-climbing that allows sideways moves with start state d

   Does it terminate? If so, where?

   Does it find the global maximum?

4.3 5-QUEENS Revisited

Now, let's revisit the 5-QUEENS problem. Our goal is to try to place 5 queens on a 5x5 chessboard with no conflict between any two queens. However, this time, there is a twist: our chessboard has a wall!



If the wall blocks the path between two queens (horizontal, vertical, or diagonal) then those two queens will not be in conflict. The wall cannot be moved, and a queen cannot pass through the wall. Below is some pseudocode to try to solve the 5-QUEENS problem with hill climbing.

```
function 5-QUEENS-HILL-CLIMBING(problem)
    current_state = problem.INITIAL-STATE
    loop do
        neighbor = state with lowest # of conflicts between queens obtained by making
            one legal chess move from current_state
        if neighbor.num_conflicts >= current_state.num_conflicts
            return current_state
        current_state = neighbor
```

Using the given starting state, pseudocode, and the rules regarding walls, answer the following questions.

1. Run 5-QUEENS-HILL-CLIMBING on the given start state until completion. Where do the queens end up? Is this a goal state?

2. If we remove the brick wall, will 5-QUEENS-HILL-CLIMBING end up in a global optima?

5 Linear Programming

1. True/False
   (a) The cost vector points in the direction of decreasing cost.
   (b) As the magnitude of c increases, the distance between the contour lines of the objective $c^T x$ increases as well.
   (c) A maximizing LP with exactly one constraint will always have a maximum objective value of ∞.
   (d) The Simplex algorithm will always visit strictly fewer vertices in a graph than using vertex enumeration.
2. The 281 TAs are throwing a big Valentine's day party for all the students. They will have a chocolate fountain and lots of fruit punch but they need help deciding how much of each to buy. Assume they can buy any fraction of an ounce. The staff has a budget of \$100. An ounce of chocolate is \$1 while an ounce of fruit punch is \$0.50. The staff wants to make sure there's enough for all the students so they need at least a total of 80 ounces of chocolate and fruit punch combined. Lastly, the 281 locker is small, and can only hold up to 120 ounces of items. Chocolate brings 3 units of happiness per ounce and fruit punch brings 1 unit of happiness per ounce, and the TAs want to maximize student happiness.
   (a) What are the variables needed to formulate this as a linear programming problem?

   (b) What are necessary constraints needed to formulate this as a linear programming problem?

   (c) Plot this linear optimization problem in the graph below and find the optimal solution. Be sure to draw in the cost vector.